Схема кода компонента Shape программы Compuct

Public Class Shape

Private Const Caption As String = "Ошибка"

ReadOnly stringManager As Resources.ResourceManager

Public Event TouchDown(sender As Shape)

Private Declare Function GetAsyncKeyState Lib "user32" (ByVal vkey As Integer) As Short

Private Const VK\_LSHIFT As Integer = &HA0

Private Const VK\_RSHIFT As Integer = &HA1

'0 - tShape, 1 - lShape, 2 - rlShape, 3 - zShape, 4 - rzShape, 5 - line, 6 - square

Private ReadOnly shapeType()() As Point = {New Point() {New Point(10, -1), New Point(10, -2), New Point(10, -3), New Point(11, -2)},

New Point() {New Point(8, -1), New Point(8, -2), New Point(9, -2), New Point(10, -2)},

New Point() {New Point(8, -1), New Point(8, -2), New Point(9, -1), New Point(10, -1)},

New Point() {New Point(8, -2), New Point(9, -1), New Point(9, -2), New Point(10, -1)},

New Point() {New Point(8, -1), New Point(9, -1), New Point(9, -2), New Point(10, -2)},

New Point() {New Point(9, -1), New Point(9, -2), New Point(9, -3), New Point(9, -4)},

New Point() {New Point(9, -1), New Point(9, -2), New Point(10, -1), New Point(10, -2)}}

Private ReadOnly rotationOffsets As New Dictionary(Of Integer, Point()())

Private ReadOnly \_xType As Integer

Private ReadOnly \_shapeColor As String

Private \_currentPoints() As Point

Private \_rotationIndex As Integer = 0

Public Sub New(xType As Integer, shapeColor As String)

Try

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Me.\_xType = xType

Me.\_shapeColor = shapeColor

Me.\_currentPoints = shapeType(xType)

rotationOffsets.Add(0, {New Point() {New Point(0, -1), New Point(1, 1), New Point(1, 1), New Point(1, 0)},

New Point() {New Point(0, 0), New Point(0, 0), New Point(0, 0), New Point(-1, -1)},

New Point() {New Point(0, 1), New Point(0, 0), New Point(0, 0), New Point(1, 2)},

New Point() {New Point(0, 0), New Point(-1, -1), New Point(-1, -1), New Point(-1, -1)}})

rotationOffsets.Add(1, {New Point() {New Point(0, -2), New Point(1, -1), New Point(0, 0), New Point(-1, 1)},

New Point() {New Point(0, 2), New Point(0, 2), New Point(1, 1), New Point(1, -1)},

New Point() {New Point(0, 0), New Point(-1, -1), New Point(-2, -2), New Point(-1, 1)},

New Point() {New Point(0, 0), New Point(0, 0), New Point(1, 1), New Point(1, -1)}})

rotationOffsets.Add(2, {New Point() {New Point(0, 0), New Point(0, 0), New Point(-1, -2), New Point(-1, -2)},

New Point() {New Point(0, -1), New Point(1, 0), New Point(2, 2), New Point(1, 1)},

New Point() {New Point(0, 1), New Point(0, 1), New Point(-1, -1), New Point(-1, -1)},

New Point() {New Point(0, 0), New Point(-1, -1), New Point(0, 1), New Point(1, 2)}})

rotationOffsets.Add(3, {New Point() {New Point(0, 1), New Point(-1, -1), New Point(0, 0), New Point(-1, -2)},

New Point() {New Point(0, -1), New Point(1, 1), New Point(0, 0), New Point(1, 2)},

New Point() {New Point(0, 1), New Point(-1, -1), New Point(0, 0), New Point(-1, -2)},

New Point() {New Point(0, -1), New Point(1, 1), New Point(0, 0), New Point(1, 2)}})

rotationOffsets.Add(4, {New Point() {New Point(0, -1), New Point(-1, -2), New Point(0, 1), New Point(-1, 0)},

New Point() {New Point(0, 1), New Point(1, 2), New Point(0, -1), New Point(1, 0)},

New Point() {New Point(0, -1), New Point(-1, -2), New Point(0, 1), New Point(-1, 0)},

New Point() {New Point(0, 1), New Point(1, 2), New Point(0, -1), New Point(1, 0)}})

rotationOffsets.Add(5, {New Point() {New Point(0, 0), New Point(1, 1), New Point(2, 2), New Point(3, 3)},

New Point() {New Point(0, 0), New Point(-1, -1), New Point(-2, -2), New Point(-3, -3)},

New Point() {New Point(0, 0), New Point(1, 1), New Point(2, 2), New Point(3, 3)},

New Point() {New Point(0, 0), New Point(-1, -1), New Point(-2, -2), New Point(-3, -3)}})

End Sub

'shape color accessor

Public ReadOnly Property ShapeColor As String

Get

Return \_shapeColor

End Get

End Property

Public ReadOnly Property CurrentPoints As Point()

Get

Return \_currentPoints

End Get

End Property

Public Property RotationIndex As Integer

Get

Return \_rotationIndex

End Get

Set(value As Integer)

\_rotationIndex = value

End Set

End Property

Public Function MoveDown(grid()() As String) As String()()

Try

Dim pts() As Point = CurrentPoints

For Each p As Point In CurrentPoints

If p.Y >= 0 Then

grid(p.Y)(p.X) = ""

End If

Next

If CanMoveBelow(CurrentPoints, grid) Then

For x As Integer = 0 To CurrentPoints.Count - 1

CurrentPoints(x).Y += 1

Dim p As Point = CurrentPoints(x)

If p.Y >= 0 Then

grid(p.Y)(p.X) = ShapeColor

End If

Next

Else

For Each p As Point In pts

If p.Y >= 0 Then

grid(p.Y)(p.X) = ShapeColor

End If

Next

End If

Return grid

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Public Function MoveLeft(grid()() As String) As String()()

Try

For Each p As Point In CurrentPoints

If p.Y >= 0 Then

grid(p.Y)(p.X) = ""

End If

Next

Dim pts() As Point = CurrentPoints

If CanMoveLeft(CurrentPoints, grid) Then

For x As Integer = 0 To CurrentPoints.Count - 1

If CurrentPoints(x).X > 0 Then

CurrentPoints(x).X -= 1

Dim p As Point = CurrentPoints(x)

grid(p.Y)(p.X) = ShapeColor

End If

Next

Else

For Each p As Point In pts

If p.Y >= 0 Then

grid(p.Y)(p.X) = ShapeColor

End If

Next

End If

Return grid

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Public Function MoveRight(grid()() As String) As String()()

Try

For Each p As Point In CurrentPoints

If p.Y >= 0 Then

grid(p.Y)(p.X) = ""

End If

Next

Dim pts() As Point = CurrentPoints

If CanMoveRight(CurrentPoints, grid) Then

For x As Integer = 0 To CurrentPoints.Count - 1

If CurrentPoints(x).X < 19 Then

CurrentPoints(x).X += 1

Dim p As Point = CurrentPoints(x)

grid(p.Y)(p.X) = ShapeColor

End If

Next

Else

For Each p As Point In pts

If p.Y >= 0 Then

grid(p.Y)(p.X) = ShapeColor

End If

Next

End If

Return grid

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Public Function RotateShape(grid()() As String) As String()()

Try

If \_xType = 6 Then Return grid

Dim shifting As Boolean = GetAsyncKeyState(VK\_LSHIFT) < 0 Or GetAsyncKeyState(VK\_RSHIFT) < 0

For Each p As Point In CurrentPoints

If p.Y >= 0 Then

grid(p.Y)(p.X) = ""

End If

Next

If RotationIndex = 4 Then RotationIndex = 0

Dim pts() As Point = DirectCast(CurrentPoints.Clone, Point())

If Not shifting Then

For x As Integer = 0 To pts.Count - 1

pts(x).Offset(rotationOffsets(\_xType)(RotationIndex)(x))

Next

If ShapeIsClear(pts, grid) Then

\_currentPoints = pts

RotationIndex += 1

End If

Else

If RotationIndex = 0 Then

RotationIndex = 3

Else

RotationIndex -= 1

End If

For x As Integer = 0 To pts.Count - 1

pts(x).Offset(NegatePoint(rotationOffsets(\_xType)(RotationIndex)(x)))

Next

If ShapeIsClear(pts, grid) Then

\_currentPoints = pts

End If

End If

For Each p As Point In CurrentPoints

If p.Y >= 0 Then

grid(p.Y)(p.X) = ShapeColor

End If

Next

Return grid

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Private Function CanMoveLeft(pts() As Point, grid()() As String) As Boolean

Try

If pts.Any(Function(p) p.Y = -1) Then Return False

For Each p As Point In pts

If p.X - 1 < 0 Then Return False

If p.Y >= 0 And (p.X > 0 And p.X < 19) Then

If Not String.IsNullOrEmpty(grid(p.Y)(p.X - 1)) Then Return False

ElseIf p.X < 0 OrElse p.X > 19 Then

Return False

End If

Next

Return True

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Private Function CanMoveRight(pts() As Point, grid()() As String) As Boolean

Try

If pts.Any(Function(p) p.Y = -1) Then Return False

For Each p As Point In pts

If p.X + 1 > 19 Then Return False

If p.Y >= 0 And (p.X > 0 And p.X < 19) Then

If Not String.IsNullOrEmpty(grid(p.Y)(p.X + 1)) Then Return False

ElseIf p.X < 0 OrElse p.X > 19 Then

Return False

End If

Next

Return True

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Private Function CanMoveBelow(pts() As Point, grid()() As String) As Boolean

Try

For Each p As Point In pts

If p.Y + 1 > 29 Then RaiseEvent TouchDown(Me) : Return False

If p.Y >= 0 Then

If Not String.IsNullOrEmpty(grid(p.Y + 1)(p.X)) Then RaiseEvent TouchDown(Me) : Return False

End If

Next

Return True

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Private Function ShapeIsClear(pts() As Point, grid()() As String) As Boolean

Try

For Each p As Point In pts

If p.Y >= 0 Then

If p.X < 0 Or p.X > 19 Then Return False

If Not String.IsNullOrEmpty(grid(p.Y)(p.X)) Then Return False

End If

Next

Return True

Catch ex As Exception

MessageBox.Show(ex.Message, stringManager.GetString(Caption, Globalization.CultureInfo.CurrentUICulture), MessageBoxButtons.OK, MessageBoxIcon.Exclamation)

End Try

Return Nothing

End Function

Private Function NegatePoint(p As Point) As Point

Return New Point(-p.X, -p.Y)

End Function

End Class